2. Count the Number of Requests by the Attacker

Create a Bash script to determine how many requests the attacker has sent, where the attacker is identified as the IP address with the highest number of requests.

Requirements:

- The script should accept a log file as input.
- It should:
  1. Identify the IP address with the most requests (assumed to be the attacker).
  2. Count the total number of requests made by that IP address.

```
┌──(oumaimaⓀhbtn-lab)-
[…/web_application_security/0x0b_web_application_fast_incident_response]
└─$  ./2-count_attack.sh
5000
```

```
awk '{print $1}' logs.txt | sort | uniq -c | sort -nr | head -n 1 | awk
'{print $1}'
```

This command extracts the first field from a file, processes it to find the most frequent occurrence, and outputs the count of that occurrence. Here's a breakdown:

Command Explanation:

1. **`awk '{print $1}' $1`**:
   - Extracts the first field (column) from the file specified as `$1` (first script argument).
2. **`sort`**:
   - Sorts the extracted values alphabetically or numerically to prepare for unique counting.
3. **`uniq -c`**:
   - Counts occurrences of each unique value. This step requires the input to be sorted.
4. **`sort -nr`**:
   - Sorts the count in numeric (`-n`) and reverse (`-r`) order so the most frequent value comes first.
5. **`head -n 1`**:
   - Extracts the top line, which corresponds to the most frequent value.
6. **`awk '{print $1}'`**:

- From the line with the most frequent value, extracts the count.

Result:

This command ultimately outputs the count of the most frequent first field in the file provided as `$1`.

Example:

Given a file `logs.txt` with the following content:

```
192.168.1.1 log1
192.168.1.2 log2
192.168.1.1 log3
192.168.1.1 log4
192.168.1.2 log5
```

Running the command:

```
awk '{print $1}' logs.txt | sort | uniq -c | sort -nr | head -n 1 | awk
'{print $1}'
```

Outputs:

```
3
```

This means `192.168.1.1` appears 3 times, the most frequent first field.